

IN THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1-17. (Cancelled)

18. (Currently amended) A system for transferring data associated with a real-time data replication system comprising:

a processor configured to use a main thread, wherein the main thread can process a synchronization command and a dynamic replication command; also configured to provide a synchronization thread, wherein the synchronization thread can process the synchronization command but not the dynamic replication command; wherein the synchronization thread is configured to process a synchronization command substantially concurrently with the processing by the main thread of a synchronization command and wherein the synchronization thread is configured to not process a synchronization command at a time when the main thread is processing a dynamic replication command; and

a memory coupled to the processor for providing the processor with instructions.

19-20. (Cancelled)

21. (Previously presented) The system of claim 18 wherein the main thread, the synchronization thread, or both process commands from a kernel cache.

22. (Previously presented) The system of claim 21 wherein the synchronization thread skips commands in the kernel cache that have been or are being processed by other threads until it

finds a synchronization command that has not yet been and is not currently being processed by another thread.

23. (Previously presented) The system of claim 21 wherein when the synchronization thread does not move ahead of the main thread unless the main thread is performing a synchronization command.

24. (Previously presented) The system of claim 21 wherein if a synchronization thread does not encounter synchronization commands in the kernel cache, the synchronization thread closes after a time interval.

25. (Previously presented) The system of claim 18 wherein the main thread does not process dynamic replication types of commands unless all synchronization threads that are executing synchronization commands are completed.

26. (New) A method for transferring data associated with a real-time data replication system comprising:

using a main thread, wherein the main thread can process a synchronization command and a dynamic replication command; and

providing a synchronization thread, wherein the synchronization thread can process the synchronization command but not the dynamic replication command;

wherein the synchronization thread is configured to process a synchronization command substantially concurrently with the processing by the main thread of a synchronization command and wherein the synchronization thread is configured to not process a synchronization command at a time when the main thread is processing a dynamic replication command.

27. (New) The method of claim 26 wherein the main thread, the synchronization thread, or both process commands from a kernel cache.

28. (New) The method of claim 27 wherein the synchronization thread skips commands in the kernel cache that have been or are being processed by other threads until it finds a synchronization command that has not yet been and is not currently being processed by another thread.

29. (New) The method of claim 27 wherein when the synchronization thread does not move ahead of the main thread unless the main thread is performing a synchronization command.

30. (New) The method of claim 27 wherein if a synchronization thread does not encounter synchronization commands in the kernel cache, the synchronization thread closes after a time interval.

31. (New) The method of claim 26 wherein the main thread does not process dynamic replication types of commands unless all synchronization threads that are executing synchronization commands are completed.

32. (New) A computer program product for transferring data associated with a real-time data replication system, the computer program product being embodied in a computer readable medium and comprising computer instructions for:

using a main thread, wherein the main thread can process a synchronization command and a dynamic replication command; and

providing a synchronization thread, wherein the synchronization thread can process the synchronization command but not the dynamic replication command;

wherein the synchronization thread is configured to process a synchronization command substantially concurrently with the processing by the main thread of a synchronization command and wherein the synchronization thread is configured to not process a synchronization command at a time when the main thread is processing a dynamic replication command.

33. (New) The computer program product recited in claim 32 wherein the main thread, the synchronization thread, or both process commands from a kernel cache.

34. (New) The computer program product recited in claim 33 wherein the synchronization thread skips commands in the kernel cache that have been or are being processed by other threads until it finds a synchronization command that has not yet been and is not currently being processed by another thread.

35. (New) The computer program product recited in claim 33 wherein when the synchronization thread does not move ahead of the main thread unless the main thread is performing a synchronization command.

36. (New) The computer program product recited in claim 33 wherein if a synchronization thread does not encounter synchronization commands in the kernel cache, the synchronization thread closes after a time interval.

37. (New) The computer program product recited in claim 32 wherein the main thread does not process dynamic replication types of commands unless all synchronization threads that are executing synchronization commands are completed.